

▼ IBM Coursera Capstone Project by Odi Dahan

```
import numpy as np # library to handle data in a vectorized manner

import pandas as pd # library for data analysis
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)

import json # library to handle JSON files

!conda install -c conda-forge geopy --yes # uncomment this line if you haven't completed the Foursquare API lab
from geopy.geocoders import Nominatim # convert an address into latitude and longitude values

import requests # library to handle requests
from pandas.io.json import json_normalize # transform JSON file into a pandas dataframe

# Matplotlib and associated plotting modules
import matplotlib.cm as cm
import matplotlib.colors as colors

# import k-means from clustering stage
from sklearn.cluster import KMeans

# for webscraping import BeautifulSoup
from bs4 import BeautifulSoup

import xml

!conda install -c conda-forge folium=0.5.0 --yes # uncomment this line if you haven't completed the Foursquare API lab
import folium # map rendering library

print('Libraries imported.')
```



Solving environment: done

```
# All requested packages already installed.
```

```
Solving environment: done

# All requested packages already installed.

Libraries imported.
```

▼ Wikipedia info: Santiago de Chile

```
wurl = requests.get("https://es.wikipedia.org/wiki/Anexo:Comunas_de_Chile_por_poblaci%C3%B3n").text
soup = BeautifulSoup(wurl, 'lxml')

table_post = soup.find('table')
fields = table_post.find_all('td')

comuna = []
region = []
provincia = []
pob2017 = []
pos_nac = []
pos_reg = []
pos_pro = []
pob2002 = []
pob1992 = []

for i in range(0, len(fields), 9):
    comuna.append(fields[i].text.strip())
    region.append(fields[i+1].text.strip())
    provincia.append(fields[i+2].text.strip())
    pob2017.append(fields[i+3].text.strip())
    pos_nac.append(fields[i+4].text.strip())
    pos_reg.append(fields[i+5].text.strip())
    pos_pro.append(fields[i+6].text.strip())
    pob2002.append(fields[i+7].text.strip())
    pob1992.append(fields[i+8].text.strip())

df = pd.DataFrame(data=[region, comuna]).transpose()
df.columns = ['Borough', 'Neighborhood']
df.head()
```

**Borough Neighborhood**

0	Metropolitana de Santiago	Puente Alto
1	Metropolitana de Santiago	Maipú
2	Metropolitana de Santiago	Santiago
3	Metropolitana de Santiago	La Florida
4	Antofagasta	Antofagasta

df.shape



(346, 2)

```
df1=df[(df.Borough == 'Metropolitana de Santiago')]  
df1.shape
```



(52, 2)

```
url = requests.get("https://es.wikipedia.org/wiki/Anexo:C%C3%B3digos_postales_de_Chile").text  
soup2 = BeautifulSoup(url,'lxml')
```

```
table = soup2.find('table')  
ff = table.find_all('td')  
  
comuna = []  
codigo = []  
  
for i in range(0, len(ff), 2):  
    comuna.append(ff[i].text.strip())  
    codigo.append(ff[i+1].text.strip())  
  
df2 = pd.DataFrame(data=[comuna,codigo]).transpose()  
df2.columns = ['Neighborhood', 'PostalCode']  
df2.head()
```



Neighborhood PostalCode

	Neighborhood	PostalCode
0	Algarrobo	2710000
1	Alhué	9650000
2	Alto Biobío	4590000
3	Alto del Carmen	1650000
4	Alto Hospicio	1130000

File data: Geo locations

```
url="https://raw.githubusercontent.com/odidahan/My-Capstone-Project/master/chile%20geo%20public.csv"
df3 = pd.read_csv(url, encoding="ISO-8859-1", sep=";", names=["Pais", "Region", "Ciudad", "Neighborhood", "Lat", "Lon"])
df3.head()
```



	Pais	Region	Ciudad	Neighborhood	Lat	Lon
0	Chile	Antofagasta	Antofagasta	Antofagasta	-23.651	-70.395
1	Chile	Antofagasta	Antofagasta	Mejillones	-23.110	-70.456
2	Chile	Antofagasta	Antofagasta	Sierra Gorda	-22.898	-69.323
3	Chile	Antofagasta	Antofagasta	Taltal	-25.410	-70.489
4	Chile	Antofagasta	El Loa	Calama	-22.474	-68.924

```
df = pd.merge(df1, df3[['Neighborhood', 'Lat', 'Lon']],
              how='left', on=['Neighborhood'])
df[39:52]
```

	Borough	Neighborhood	Lat	Lon
39	Metropolitana de Santiago	Talagante	-33.667	-70.931
40	Metropolitana de Santiago	Paine	-33.812	-70.723
41	Metropolitana de Santiago	Padre Hurtado	-33.576	-70.800
42	Metropolitana de Santiago	Isla de Maipo	-33.754	-70.886
43	Metropolitana de Santiago	El Monte	-33.684	-71.017
44	Metropolitana de Santiago	Curacaví	-33.399	-71.137
45	Metropolitana de Santiago	Pirque	-33.650	-70.564
46	Metropolitana de Santiago	Colera do Tanco	-33.628	-70.785

	Municipio/Comuna	Localidad	Lat	Lon
47	Metropolitana de Santiago	Tiltil	-33.085	-70.925
48	Metropolitana de Santiago	San José de Maipo	-33.644	-70.353
49	Metropolitana de Santiago	María Pinto	-33.515	-71.119
50	Metropolitana de Santiago	San Pedro	-33.900	-71.467
51	Metropolitana de Santiago	Alhué	-34.033	-71.100

```
dff.shape
```

(52, 4)

```
df1 = dff.copy()  
df1.head(10)
```

	Borough	Neighborhood	Lat	Lon
0	Metropolitana de Santiago	Puente Alto	-33.616	-70.570
1	Metropolitana de Santiago	Maipú	-33.490	-70.788
2	Metropolitana de Santiago	Santiago	-33.425	-70.566
3	Metropolitana de Santiago	La Florida	-33.525	-70.538
4	Metropolitana de Santiago	San Bernardo	-33.582	-70.687
5	Metropolitana de Santiago	Las Condes	-33.400	-70.503
6	Metropolitana de Santiago	Peñalolén	-33.482	-70.538
7	Metropolitana de Santiago	Pudahuel	-33.411	-70.836
8	Metropolitana de Santiago	Quilicura	-33.361	-70.729
9	Metropolitana de Santiago	Ñuñoa	-33.454	-70.604

▼ Map of Santiago de Chile

```
address = 'Santiago, Chile'

geolocator = Nominatim(user_agent="capstoneProject")
location = geolocator.geocode(address, timeout=60, exactly_one=True)
latitude = location.latitude
longitude = location.longitude
print('The decimal coordinates of Santiago are {}, {}'.format(latitude, longitude))
```

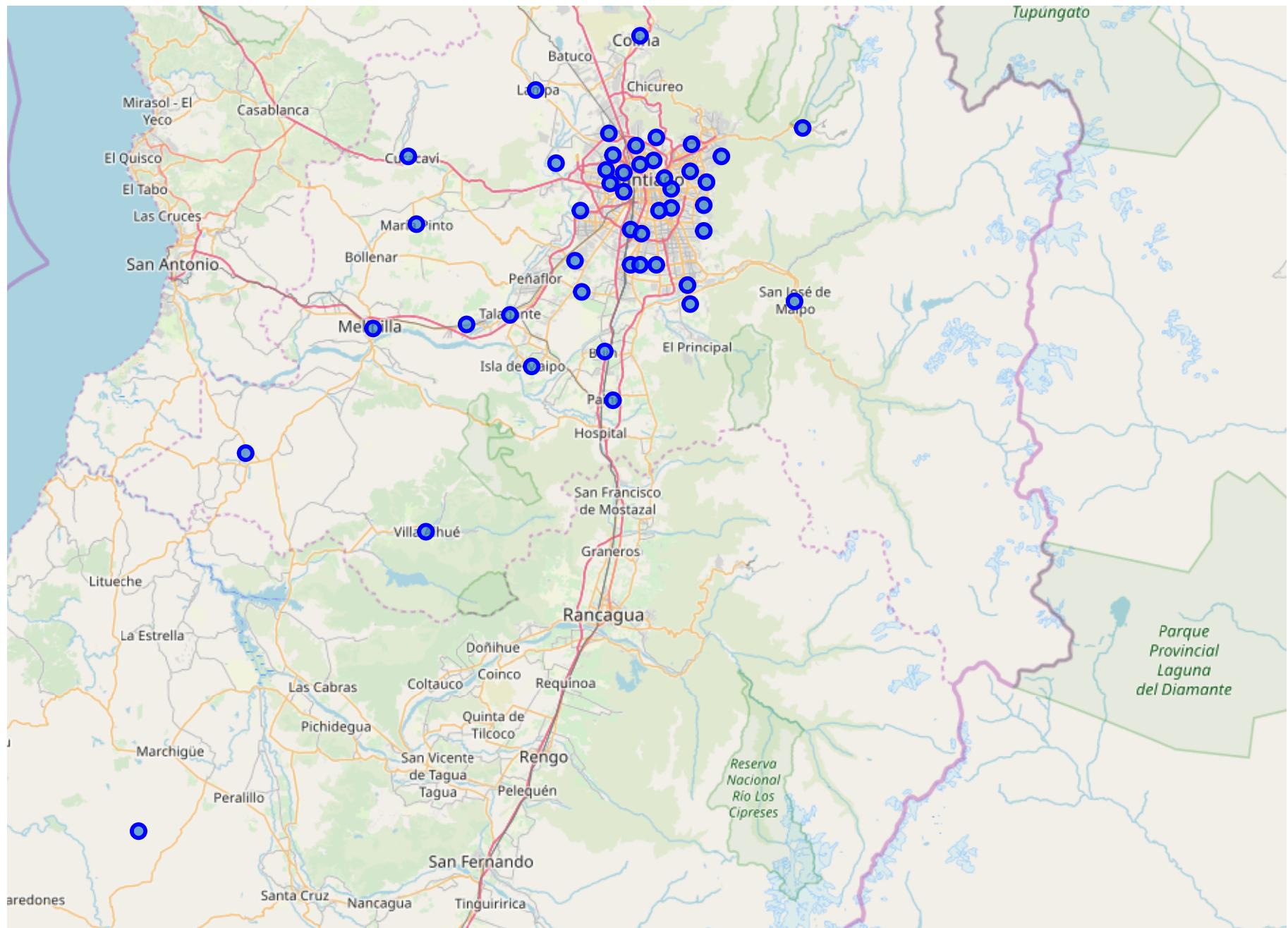
 The decimal coordinates of Santiago are -33.4377968, -70.6504451.

```
map_stgo = folium.Map(location=[latitude, longitude], zoom_start=12)

# add markers to map
for lat, lng, local in zip(dff1['Lat'], dff1['Lon'], dff1['Neighborhood']):
    label = '{}'.format(local)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='blue',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7).add_to(map_stgo)
```

```
map_stgo
```







```
import urllib
def getNearbyVenues(names, latitudes, longitudes, radius=5000, categoryIds=''):
    try:
        venues_list=[]
        for name, lat, lng in zip(names, latitudes, longitudes):
            #print(name)

            # create the API request URL
            url = 'https://api.foursquare.com/v2/venues/search?&client_id={}&client_secret={}&v={}&ll={},{}&radius={}&limit={}

            if (categoryIds != ''):
                url = url + '&categoryId={}'
                url = url.format(categoryIds)

            # make the GET request
            response = requests.get(url).json()
            results = response[ "response"]['venues']

            # return only relevant information for each nearby venue
            for v in results:
                success = False
                try:
                    category = v[ 'categories'][0][ 'name']
                    success = True
                except:
```

```
except:  
    pass  
  
if success:  
    venues_list.append([ (   
        name,  
        lat,  
        lng,  
        v[ 'name' ],  
        v[ 'location' ][ 'lat' ],  
        v[ 'location' ][ 'lng' ],  
        v[ 'categories' ][ 0 ][ 'name' ]  
    ) ])   
  
nearby_venues = pd.DataFrame([item for venue_list in venues_list for item in venue_list])  
nearby_venues.columns = [ 'Localidad',  
    'Localidad Latitude',  
    'Localidad Longitude',  
    'Venue',  
    'Venue Latitude',  
    'Venue Longitude',  
    'Venue Category' ]  
  
except:  
    print(url)  
    print(response)  
    print(results)  
    print(nearby_venues)  
  
return(nearby_venues)
```

▼ Foursquare venues

```
LIMIT = 500  
radius = 5000  
CLIENT_ID = '2300WU0UYZEZYCLUQ1VCEX1GT4RAOZEPHJSVCY20D2G1PYNU'  
CLIENT_SECRET = 'OGGYNC042YPWHH0YDSQIAUPA0QR3MLKHGL5UOUSOOYU2MLB'  
VERSION = '20190908'
```

```
#https://developer.foursquare.com/docs/resources/categories  
#Sushi = 4bf58dd8d48988d1d2941735  
stgo_venues_sushi = getNearbyVenues(names=dff1[ 'Neighborhood' ], latitudes=dff1[ 'Lat' ], longitudes=dff1[ 'Lon' ], radius=1000, ca  
stgo_venues_sushi.head()
```

```
~~~_-----_-----\`
```

	Localidad	Localidad	Latitude	Localidad	Longitude	Venue	Venue	Latitude	Venue	Longitude	Venue	Cat
0	Puente Alto		-33.616		-70.57	Sushi Han' El Delivery		-33.610953		-70.572730	Sushi Re	
1	Puente Alto		-33.616		-70.57		Sushi Illadi	-33.612373		-70.574906	Sushi Re	
2	Puente Alto		-33.616		-70.57		Mazushi	-33.612084		-70.576012	Sushi Re	
3	Puente Alto		-33.616		-70.57		Batak's	-33.609552		-70.577479	Sushi Re	
4	Puente Alto		-33.616		-70.57		Sushi Bar Otai	-33.609399		-70.575004	Sushi Re	

stgo_venues_sushi.shape

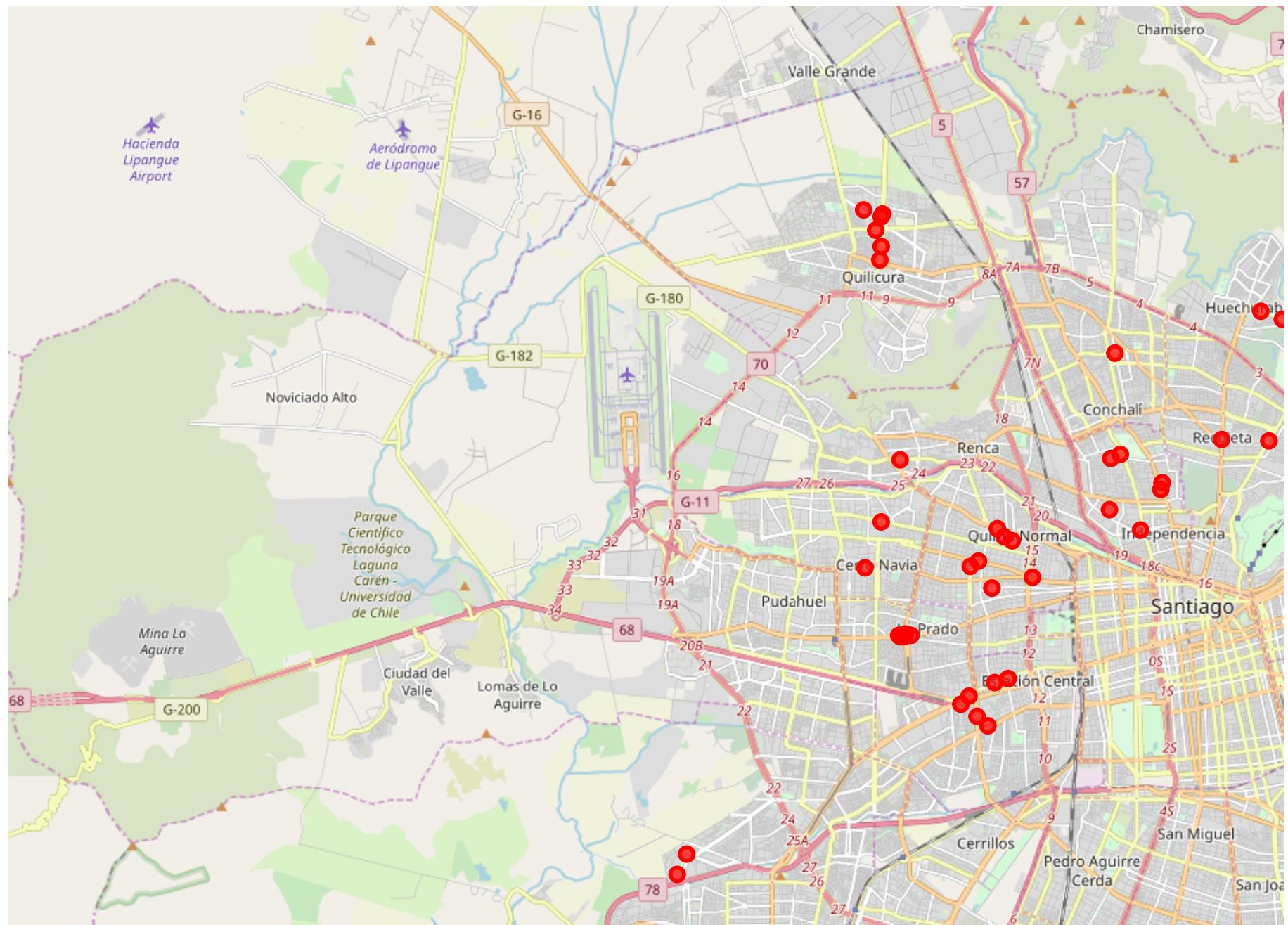
```
(184, 7)
```

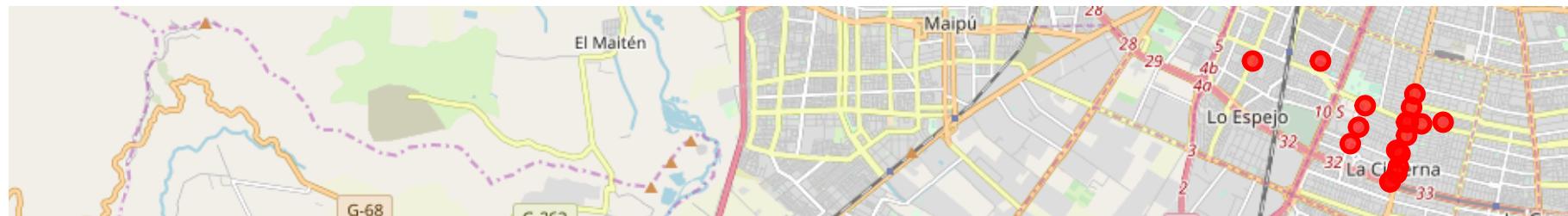
```
def addToMap(df, color, existingMap):
    for lat, lng, local, venue, venueCat in zip(df['Venue Latitude'], df['Venue Longitude'], df['Localidad'], df['Venue'], df['Category']):
        label = '{} ({}) - {}'.format(venue, venueCat, local)
        label = folium.Popup(label, parse_html=True)
        folium.CircleMarker(
            [lat, lng],
            radius=5,
            popup=label,
            color=color,
            fill=True,
            fill_color=color,
            fill_opacity=0.7).add_to(existingMap)
```

map_stgo_sushi = folium.Map(location=[latitude, longitude], zoom_start=12)

addToMap(stgo_venues_sushi, 'red', map_stgo_sushi)







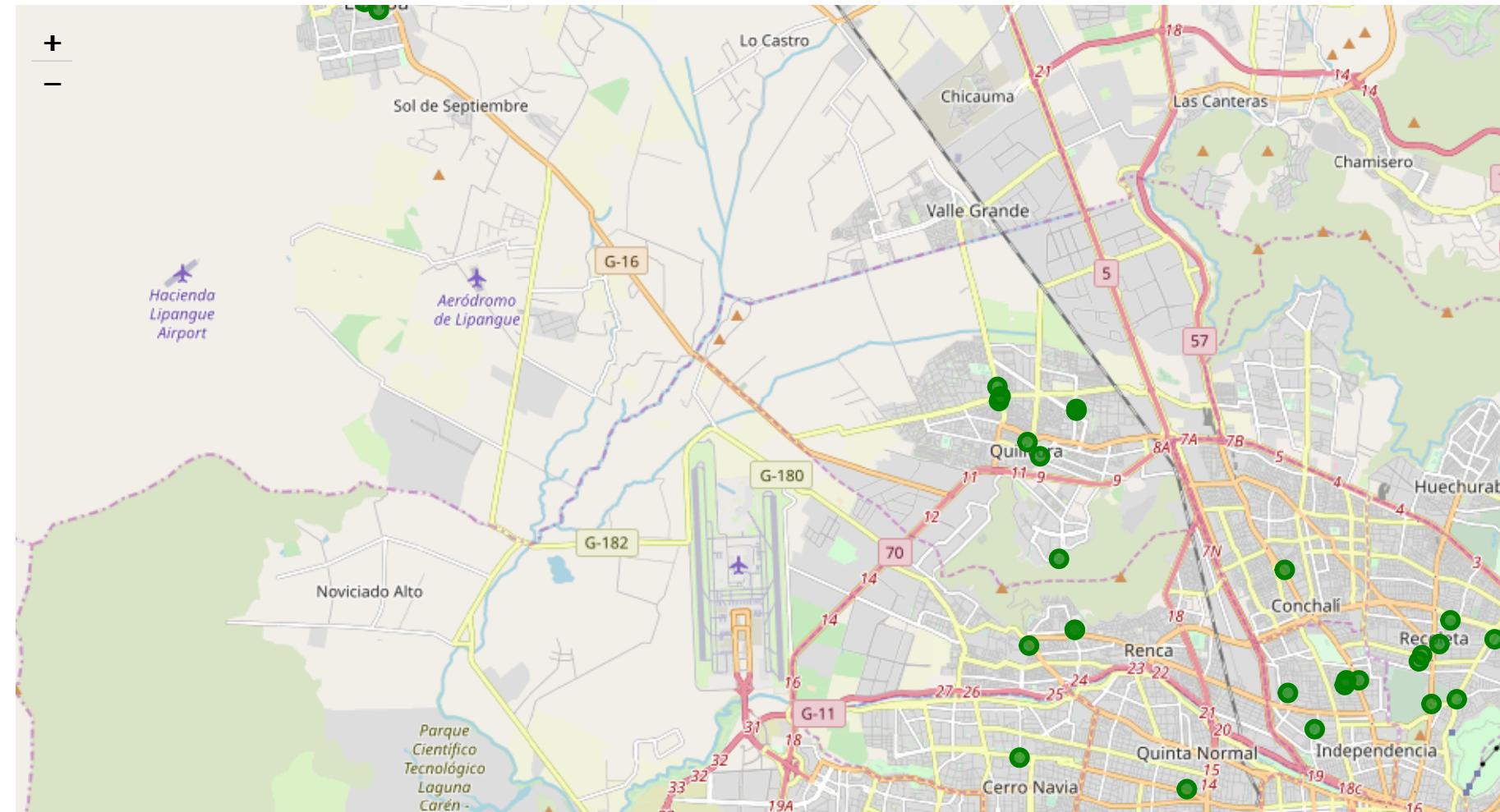
```
#Highschools = 4bf58dd8d48988d13d941735
stgo_venues_highschools = getNearbyVenues(names=dff1['Neighborhood'], latitudes=dff1['Lat'], longitudes=dff1['Lon'], radius=10
stgo_venues_highschools.head()
```

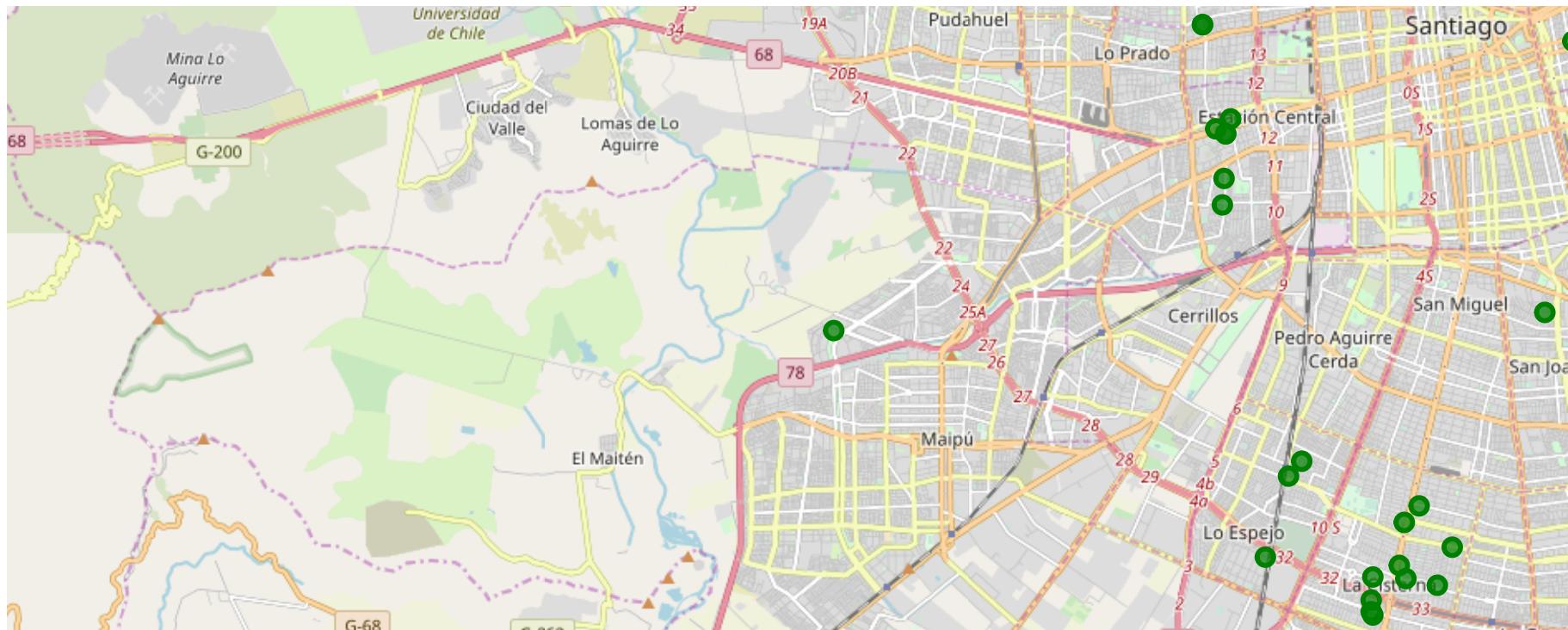
	Localidad	Localidad	Latitude	Localidad	Longitude	Venue	Venue	Latitude	Venue	Lon
0	Puente Alto		-33.616		-70.570	Complejo Eduacional Consolidada		-33.611524		-70
1	Puente Alto		-33.616		-70.570		Colegio Tacora	-33.609515		-70
2	Puente Alto		-33.616		-70.570		Colegio Nidal	-33.611392		-70
3	Puente Alto		-33.616		-70.570	Colegio Polivalente Domingo Matte Mesias		-33.608298		-70
4	Maipú		-33.490		-70.788		Colegio Alicante El Rosal	-33.490642		-70

stgo_venues_highschools.shape

(137, 7)

```
map_stgo_highschools = folium.Map(location=[latitude, longitude], zoom_start=12)
addToMap(stgo_venues_highschools, 'green', map_stgo_highschools)
map_stgo_highschools
```





```
stgo_venues_uni = getNearbyVenues(names=dff1['Neighborhood'], latitudes=dff1['Lat'], longitudes=dff1['Lon'], radius=1000, categories='College')
stgo_venues_uni.head()
```



	Localidad	Localidad	Latitude	Localidad	Longitude
0	Santiago		-33.425		-70.566

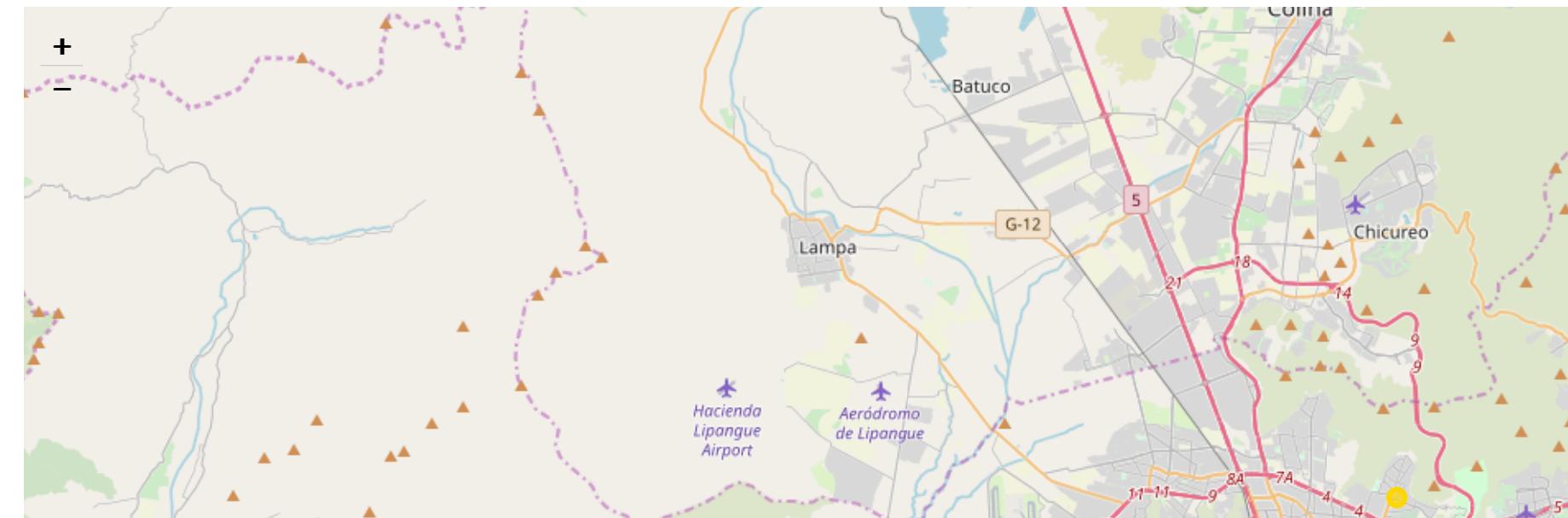
	Venue	Venue	Latitude	Venue	Longitude	Venue
0	Sala de Lenguaje		-33.425443		-70.569583	College

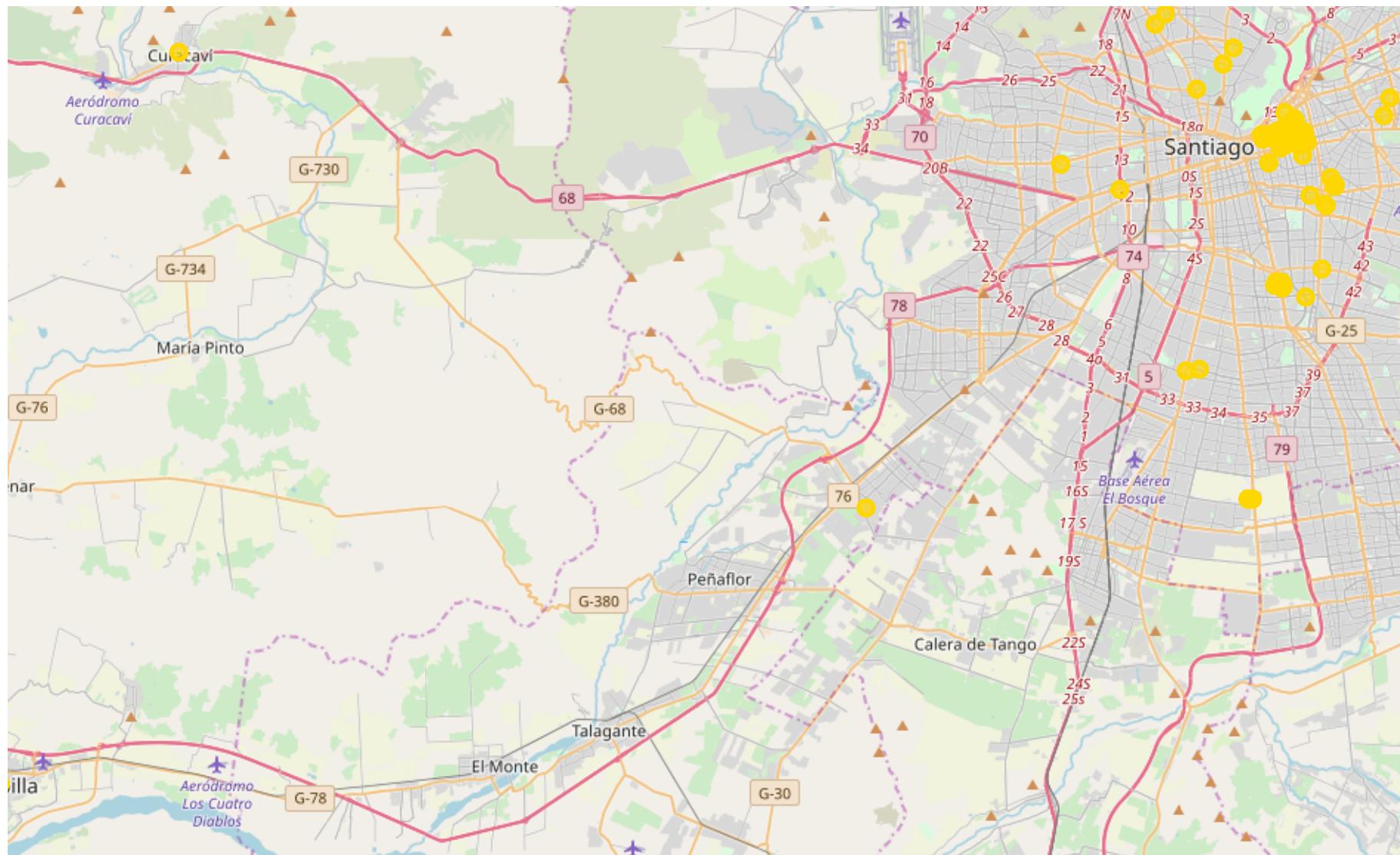
1	Santiago	-33.425	-70.566	Aula 3- Incacea	-33.417995	-70.556946
2	Santiago	-33.425	-70.566	Casa Estudios	-33.418552	-70.566596
3	Santiago	-33.425	-70.566	Sala de sonido-Incacea	-33.420655	-70.557696
4	Las Condes	-33.400	-70.503	Duoc UC I San Carlos II	-33.400133	-70.506968

stgo_venues_uni.shape

(138, 7)

```
map_stgo_universities = folium.Map(location=[latitude, longitude], zoom_start=12)
addToMap(stgo_venues_uni, 'gold', map_stgo_universities)
map_stgo_universities
```





```
stgo_venues_office = getNearbyVenues(names=dff1['Neighborhood'], latitudes=dff1['Lat'], longitudes=dff1['Lon'], radius=1000, c  
stgo_venues_office.head()
```



	Localidad	Localidad	Latitude	Localidad	Longitude	Venue	Venue	Latitude	Venue	Longitude
0	Puente Alto		-33.616		-70.57	Molino Puente Alto		-33.612333		-70.573923
1	Puente Alto		-33.616		-70.57		Vidaintegra	-33.609392		-70.574928
2	Puente Alto		-33.616		-70.57	CMPC - Papeles Cordillera		-33.610262		-70.563704
3	Puente Alto		-33.616		-70.57	Fundacion Vivir Libres		-33.612088		-70.569455
4	Puente Alto		-33.616		-70.57	Frigorifico Y Faenadora Cordillera		-33.622742		-70.568747

```
defaddColumn(startDf, columnTitle, dataDf):  
    grouped = dataDf.groupby('Localidad').count()  
  
    for n in startDf['Localidad']:  
        try:  
            startDf.loc[startDf['Localidad'] == n, columnTitle] = grouped.loc[n, 'Venue']  
        except:  
            startDf.loc[startDf['Localidad'] == n, columnTitle] = 0
```

```
df_data = dff1.copy()
df_data.rename(columns={'Neighborhood': 'Localidad'}, inplace=True)
addColumn(df_data, 'Sushi', stgo_venues_sushi)
addColumn(df_data, 'High Schools', stgo_venues_highschools)
addColumn(df_data, 'Universities', stgo_venues_uni)
addColumn(df_data, 'Offices', stgo_venues_office)
df_data.head()
```

	Borough	Localidad	Lat	Lon	Sushi	High Schools	Universities	Offices
0	Metropolitana de Santiago	Puente Alto	-33.616	-70.570	5.0	4.0	0.0	46.0
1	Metropolitana de Santiago	Maipú	-33.490	-70.788	2.0	1.0	0.0	16.0
2	Metropolitana de Santiago	Santiago	-33.425	-70.566	6.0	9.0	4.0	49.0
3	Metropolitana de Santiago	La Florida	-33.525	-70.538	0.0	0.0	0.0	18.0
4	Metropolitana de Santiago	San Bernardo	-33.582	-70.687	3.0	6.0	0.0	49.0

```
# negative weight
weight_sushi = -1

# positive weight, ascending
weight_schools = 1
weight_uni = 2
weight_offices = 3
```

```
df_weighted = df_data[['Localidad']].copy()
```

```
df_weighted['Score'] = df_data['Sushi'] * weight_sushi + df_data['High Schools'] * weight_schools + df_data['Universities'] *
df_weighted = df_weighted.sort_values(by=['Score'], ascending=False)
df_weighted[39:52]
```

**Localidad Score**

41	Padre Hurtado	27.0
46	Calera de Tango	24.0
40	Paine	21.0
7	Pudahuel	12.0
50	San Pedro	6.0
45	Pirque	3.0
20	La Granja	3.0
24	Lo Barnechea	0.0
38	Cerrillos	0.0
37	San Ramón	0.0
11	El Bosque	0.0
34	Peñaflor	0.0
23	San Miguel	0.0

```
map_stgo_result = folium.Map(location=[latitude, longitude], zoom_start=15)

stgo_win = dff1[dff1['Neighborhood'] == 'Providencia']

for lat, lng, local in zip(stgo_win['Lat'], stgo_win['Lon'], stgo_win['Neighborhood']):
    label = '{}'.format(local)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='blue',
        fill=True,
        fill_color='blue',
```

```
fill_opacity=0.7).add_to(map_stgo_result)
```

```
addToMap(stgo_venues_sushi[stgo_venues_sushi['Localidad'] == 'Providencia'], 'red', map_stgo_result)
addToMap(stgo_venues_highschools[stgo_venues_highschools['Localidad'] == 'Providencia'], 'green', map_stgo_result)
addToMap(stgo_venues_uni[stgo_venues_uni['Localidad'] == 'Providencia'], 'gold', map_stgo_result)
addToMap(stgo_venues_office[stgo_venues_office['Localidad'] == 'Providencia'], 'fuchsia', map_stgo_result)
```

```
map_stgo_result
```

